notes

1 Foundations

1.1 The Role of Algorithms in Computing

1.1.1 Algorithms

1.1.2 Algorithms as a technology

1.1.3 Problems

1.2 Getting Started

1.2.1 Insertion sort

Input:A sequence of n numbers $(a_1, a_2, \cdots, a_n)$.

Output:A permutation $(a'_1, a'_2, \cdots, a'_n)$ of the input sequence such that $a'_1 \leq a'_2 \leq \cdots \leq a'_n$.

```
INSERTION-SORT(A)
for j ← 2 to length[A]
    do key ← A[j]
       ▷ Insert A[j] into the sorted sequence A[1 .. j − 1].
       i ← j − 1
       while i > 0 and A[i] > key
           do A[i + 1] ← A[i]
              i ← i − 1
       A[i + 1] ← key
```

loop invariant:We use loop invariants to help us understand why an algorithm is correct.We must show three things about a loop invariant:

Initialization:It is true prior to the first iteration of the loop.

Maintenance:If it is true before an iteration of the loop,it remains true before the next iteration.

Termination:When the loop terminates,the invariant gives a useful property that helps show the algorithm is correct.

1.2.2 Analyzing algorithms

1.2.3 Designing algorithms

1.2.4 Problems